Application. No: 10/679,870	§
Filed: October 6, 2003	§

Inventor(s):
    Patrick Williams, David
    Rohacek
	§
	 §
	 §
	 §
	 §

Title: PARAMETER
       ASSISTANT FOR
       AUTOMATICALLY
       SPECIFYING
       PARAMETER VALUES
       FOR A SOFTWARE
       PROGRAM
	§
	 §
	 §
	 §
	 §
	 §
	 §

Examiner: Yigdall, Michael J.	§
Group/Art Unit: 2192	

Atty. Dkt. No: 5150-77600

REPLY BRIEF

Box: Reply Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir/Madam:

     Further to the Examiner's Answer of September 09, 2009, Appellant presents this Reply Brief, and respectfully requests that this Reply Brief be considered by the Board of Patent Appeals and Interferences.

REMARKS

Arguments

Appellant respectfully submits that the Examiner has improperly interpreted key terms in the claims contrary to not only their commonly understood meanings, but also contrary to the Specification and Appellant's clearly stated intent, as explained below.

Regarding Examiner's further arguments directed to the section 103 rejection of claims 1, 9, 11, 13, 20-24, 26, 27, 31 and 32 over U.S. Patent No. 6,026,233 to Shulman et al. (hereinafter "Shulman") in view of U.S Patent No. 5,784,275 to Sojoodi et al. (hereinafter "Sojoodi"), Appellant presents the following arguments:

Independent Claims 1, 24, 26 and 27

Regarding the limitation of claim 1:

> programmatically determine one or more valid parameter values for the first parameter of the first function call by invoking software for a measurement device in order to determine one or more <u>hardware resources</u> of the measurement device, wherein each of the one or more valid parameter values represents a respective hardware resource of the one or more hardware resources,

the Examiner admits that the cited VISA classes are not hardware resources, but then argues that they represent hardware resources. This is incorrect.

As explained in the Appeal Brief, Sojoodi specifically states:

> The VISA library 52 comprises executable functions which are called by the VI 50 to perform various operations in order to control the instrument 54. Examples of the executable functions are viOpen(), viClose(), viRead(), viwrite(), viSetAttribute(), viGetAttribute(), viPeek(), viPoke(), viAssertTrigger(), viClear(), and viReadSTB(). These functions perform operations to control the instrument 54 as defined in the VISA specification in Appendix C. VISA resources comprise the operations, along with numerous attributes defined in the VISA specification. <u>The VISA nodes 66 and VISA session control 74 comprise various classes and objects, according to the notion of classes and objects in the art of object-oriented programming. These classes and objects correspond to the resources in the VISA I/O Control library 52</u> as will be explained in more detail below. (*Col. 12, lines 49-64*)

In other words, the cited classes and objects represent *software* resources in a software library that in turn are used to *control* hardware devices. As the citation makes quite clear, the cited resources are VISA I/O Control library (software) items, *not* hardware resources. As one of skill in the art would readily understand, a class that corresponds to or represents *a software resource* that is used to *interface* with a hardware resource does not represent the hardware resource itself. Said another way, determining software objects or classes that implement control interfaces *for* hardware devices is *not* equivalent to determining the hardware resources themselves. Appellant notes, for example, that in Sojoodi's system, determining the VISA I/O interface classes for interfacing with hardware devices does not include determining the hardware devices (hardware resources) themselves, whereas claim 1 specifically recites invoking software *to determine one or more* <u>hardware resources</u> *of the measurement device*. Sojoodi is quite clear that the cited VISA classes correspond to software functions for use with *possible interface types of the instrument*, i.e., to different software interfaces for communicating with or controlling the instrument, not to respective hardware resources.

The Examiner asserts that a hardware I/O interface is a hardware resource. Appellant asserts that the software constructs in Sojoodi are NOT hardware I/O interfaces (assuming that the Examiner uses the term "hardware I/O interface" to refer to hardware). The software objects in Sojoodi are not hardware resources; rather, as explained above, and as the above quote from Sojoodi makes clear, the software interfaces of Sojoodi are software classes and objects used to interface with hardware devices, not hardware resources. *Hardware resources are not software objects, nor are software interfaces for hardware resources themselves hardware resources.* Thus, contrary to the Examiner's assertion, "a program editor querying the object manager for a list of such classes" does not, in fact, suggest "invoking software for a measurement device in order to determine one or more hardware resources for the measurement device".

Thus, the cited art fails to disclose this feature.

Regarding the limitation of claim 1:

receive user input to the graphical user interface to <u>select a first</u> parameter value from the one or more valid parameter values, wherein the first parameter value represents a first hardware resource of the measurement device; and

automatically modify the first function call displayed in the source code of the software program by including the first parameter value in the first function call in response to the user input selecting the first parameter value, wherein automatically including the first parameter value in the first function call aids a user in modifying the first function call to reference the first hardware resource of the measurement device,

particularly, regarding the Examiner's interpretation "the first function call [would be] modified by including a first parameter value representing a particular VISA class in the first function call", Appellant respectfully notes that a parameter value representing a software class that represents a software interface for an instrument is clearly not equivalent to a parameter value representing a hardware resource, since a software class is not a hardware resource, and a software interface for an instrument is not itself a hardware resource of that instrument.

The Examiner admits that Shulman fails to teach that the parameters represent hardware resources, but then argues that Shulman combined with Sojoodi teaches this feature. However, as shown above, Sojoodi also fails to disclose parameters that represent hardware resources. Thus, even in combination, the cited art does not, and cannot, disclose these claimed features.

The Examiner asserts that "Appellant's invention, as claimed, is directed to an embodiment of the tool described in Shulman wherein the 'first function call' references a hardware resource of a measurement device". This is incorrect. Appellant respectfully submits that it is improper for the Examiner to simply insert key features of Appellant's claim into the Examiner's alleged construction when the cited references neither teach nor suggest the features. In other words, the Examiner has added features not disclosed in the cited references to the alleged combination based on Appellant's claims, i.e., using Appellant's claims as a blueprint, which is improper.

The Examiner further argues that Sojoodi's function calls to read or write data from or to an instrument somehow teaches these claimed features, noting that "the VISA classes noted above encapsulate these functions to control the instrument or measurement

device from the programming environment". This is incorrect. As one of ordinary skill in the art would readily understand, a parameter that represents a class that encapsulates functions for interfacing *with* a device is not equivalent to a parameter that represents the device. For example, a pencil used to write data to a piece of paper does not represent the paper, but rather, is simply a tool for interacting with the paper. Similarly, a parameter that represents such a pencil does not represent the paper.

The Examiner then continues to argue that "Sojoodi describes a tool that, like the teachings of Shulman, is also comparable to Appellant's parameter assistant", citing the description of Figure 28, which states "The Select Item pull-right menu only displays attributes for the user to select which are valid for the current VISA class of the attribute node". Appellant again notes that attributes for a VISA class (which corresponds to software objects for communicating with a hardware device) are not parameter values that represent hardware resources. Thus, the Examiner's assertion regarding the equivalence of Sojoodi's tool to the functionality recited in claim 1 is incorrect.

Thus, the cited art also fails to disclose these features of claim 1.

Appellant thus respectfully submits that independent claim 1 is patentably distinct over the cited art for at least the reasons set forth above.

Inasmuch as the independent claims 26 and 27 recite limitations similar or identical to those discussed above with reference to claim 1, Appellant respectfully submits that these claims are also patentably distinct over the cited art.

The independent claim 24 also recites similar limitations as claim 1, except that instead of a first function call it recites a first method call. As discussed above with respect to claim 1, Appellant respectfully submits that the Examiner's equation of software classes with hardware resources is erroneous. Appellant thus respectfully submits that claim 24 is also patentably distinct over the cited art.

Independent Claim 31

Regarding the limitation of claim 31:

programmatically determine one or more valid parameter values for the first input parameter of the first node by invoking software for a measurement device in order to determine one or more <u>hardware</u> resources of the measurement device, wherein each of the one or more valid parameter values

represents a respective hardware resource of the one or more hardware resources;,

Appellant respectfully submits that the Examiner's equating of software classes with hardware resources is erroneous, as explained at length above. The position of the Examiner that "the VISA classes described in Sojoodi are <u>representations</u> of hardware resources" is incorrect. Rather, per the above, these classes represent or implement software functions for interfacing or controlling hardware devices, which is quite different. The arguments provided above apply to this feature of claim 31, and so are not repeated here.

Thus, the cited art fails to teach or suggest these features of claim 31, and so claim 31 is patentably distinct over the cited art for at least these reasons.

Regarding the limitations:

> display a block diagram of a graphical program, wherein the block diagram includes a plurality of interconnected nodes visually indicating functionality of the graphical program, wherein the block diagram can be compiled into executable code, wherein the plurality of interconnected nodes includes a first node that takes a first input parameter;

and

> automatically configure the first node with the first parameter value in response to the user input selecting the first parameter value, wherein automatically configuring the first node with the first parameter value comprises automatically updating the displayed block diagram to visually indicate that the first node receives the first parameter value as input.

the Examiner argues that Shulman's modification of a function call in a text-based programming language combined with Sojoodi's graphical program function nodes somehow renders these specific claim features obvious, asserting that these two references would have suggested these particular limitations to one of ordinary skill in the art. Appellant submits that this is simply speculation, and respectfully reminds the Examiner that to establish a prima facie obviousness of a claimed invention, all claim limitations must be taught or suggested by the prior art. In re Royka, 490 F.2d 981, 180 U.S.P.Q. 580 (C.C.P.A. 1974), MPEP 2143.03.

Modifying a function call written in a text-based programming language is *not* the same as automatically configuring a node in a block diagram of a graphical program with a parameter value, and so Shulman and Sojoodi, taken either singly or in combination, do not teach the recited limitation of automatically configuring the first node with the first parameter value in response to the user input selecting the first parameter value, wherein automatically configuring the first node with the first parameter value comprises automatically updating the displayed block diagram to visually indicate that the first node receives the first parameter value as input.

Similarly, Sojoodi, taken either singly or in combination with Shulman, does not teach "automatically configure the first node with the first parameter value in response to the user input selecting the first parameter value, wherein automatically configuring the first node with the first parameter value comprises automatically updating the displayed block diagram to visually indicate that the first node receives the first parameter value as input".

The Examiner's hindsight analysis using Appellant's claims as a blueprint is improper, as is the Examiner's unfounded construction of claim features that are not actually disclosed in the references, and thus a prima facie case of obviousness is not supported, given that the references in combination do not disclose these features.

Claim 32

Claim 32 recites the further limitations of:

wherein automatically configuring the first node with the first parameter value comprises automatically wiring the first parameter value to an input terminal of the first node;
wherein updating the block diagram comprises displaying a wire connecting the first parameter value to the input terminal of the first node.

The arguments presented above with respect to claim 31 also apply generally to claim 32, since the references fail to describe these specific features. Again, the Examiner has improperly attempted to construct features not actually disclosed in the references, based on speculations made in hindsight.

Claims 20 and 21

Claim 20 recites the further limitations of: "wherein the source code is displayed in a first window," and "wherein said displaying the graphical user interface comprises displaying the graphical user interface in a separate window apart from the first window".

Claim 21 recites the further limitations of: "wherein the source code is displayed in a first portion of a first window," and "wherein said displaying the graphical user interface comprises displaying the graphical user interface in a second portion of the first window".

Again, Appellant submits that the Examiner's two interpretations of Figure 8 are inconsistent with each other, and thus Figure 8 does not, and cannot, disclose both of these mutually exclusive features. Moreover, the Examiner's argument ignores or misrepresents the claim language, arguing that according to one interpretation, "the source code is displayed in a 'first portion' of the edit display screen 700 (i.e., at 732) and the selection menu assist window 850 is displayed in a 'second portion' of the edit display screen 700". As one of skill in the art would readily understand, "portions" of a display screen are not interchangeable with "windows", and so, since Shulman's assist window is a separate window from the window in which Shulman's source code is displayed, Shulman does not, and cannot, teach the limitation in claim 21 of "wherein said displaying the graphical user interface comprises displaying the graphical user interface in a second portion of the <u>first window</u>". Accordingly Appellant respectfully submits that claim 21 is separately patentable over the cited art.

Section 103 Rejections (Shulman in view of Sojoodi and Molinari)

Regarding claims 33 and 34, and claim 12, the Examiner continues to assert equivalence between Sojoodi's VISA software classes/objects that correspond to software interfaces for communicating with hardware devices, with hardware resources, which, as explained at length above, is incorrect.

While Appellant disagrees with the Examiner's continued arguments regarding these claims, for brevity, Appellant has not specifically responded to them, but submits that the cited references fail to teach or suggest all the particular features of these claims, as well as those of their respective independent claims.

CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of the claims was erroneous, and reversal of the decision is respectfully requested.

If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above-referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. The Commissioner is hereby authorized to charge any fees which may be required or credit any overpayment to Meyertons, Hood, Kivlin, Kowert & Goetzel P.C., Deposit Account No. 50-1505/5150-77600/JCH.

Respectfully submitted,

/Jeffrey C. Hood/
Jeffrey C. Hood, Reg. #35198
ATTORNEY FOR APPLICANT(S)

Meyertons Hood Kivlin Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: 2010-01-11   JCH/MSW